REMARKS/ARGUMENTS

Claims 1, 2, 5-10, 12-16, 18-22 and 24-28 are pending and rejected. Claims 3-4, 11, 17 and 23 were previously cancelled.

Claims 1, 2, 5-10, 12-16, 18-22 and 24-28 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Fujii et al., (hereinafter "Fujii"), (US Pat. Application Publication 2004/0015888, in view of Dowling, (hereinafter "Dowling"), (US 6,363,475).

With regard to § 103 rejection of claim 1, Applicants submit the cited references do not teach or suggest at least an execution unit to execute a first thread and a second thread in parallel; and a multi-thread scheduler coupled to said first instruction fetch unit, said second instruction fetch unit, and said execution unit, wherein said multi-thread scheduler is to determine the width of said execution unit (*e.g.*, as described in claim 1).

First, Applicants agree with the Office Action's indication Fujii fails to disclose at least the multi-thread scheduler to determine a width of an execution unit. *See* Office Action dated 12/21/2007, paragraph 6. To make up for the deficiencies of Fujii, the Office Action asserts Dowling discloses the relevant limitations, citing the Abstract and column 5, lines 53-63. *See* id. Applicants disagree.

The Abstract of Dowling states that the reference is directed to exploiting parallelism inherent in a VLIW processor by providing new instruction level mechanisms to separate processor execution into parallel threads. In Dowling, a first program and a second program execute concurrently such that the second program executes using

resources and cycles that would have been wasted by the second program. *See* cited Abstract.

As shown by the Abstract, Dowling is directed to the use of program level generally parallelism in the program level directed to VLIW processors. The Abstract does not teach or suggest a multi-scheduler as described in claimed embodiments of the present application anywhere, and does not teach or suggest the use of such a multi-scheduler to determine the width of an execution unit.

The second section cited section, column 5, lines 53-63, state:

> The method includes the step of determining which instructions in the first prefetch buffer are ready to dispatch in parallel in a given cycle. The method dispatches all instructions determined to be ready to dispatch from the first prefetch buffer for which hardware resources are available. The method determines which instructions in the second prefetch buffer are ready to dispatch in parallel in said given cycle. The method dispatches one or more instructions from the second prefetch buffer using the hardware resources not already in use by the instructions in the first prefetch buffer.

The cited section is describes aspects similar to those described in the Abstract. In particular, the method described, after dispatching all ready instructions from the first prefetch buffer, determines which instructions in the second prefetch buffer are ready to dispatch in parallel. The second instructions from the second prefetch buffer are dispatched to hardware resources not in use by the instructions in the first prefetch buffer.

Similar to the Abstract discussed above, this cited section is directed to parallel processing among various hardware resources, wherein the resources not being utilized by the instructions from the first prefetch buffer are utilized to process instructions from the second prefetch buffer. Program level parallel processing is not the same as utilizing

a multi-thread scheduler, wherein the multi-thread scheduler determines the width of an execution unit (*e.g.*, as described in claim 1). There is no teaching, suggestion, or even implication that a width of an execution unit is considered in performing the methods described in Dowling – as discussed above, it is directed merely to allocation of hardware resources in parallel processing on a program level. Indeed, the cited sections do not address a multi-thread scheduler, or the use of such to determine the width of an execution unit at all.

In order to support a proper rejection of claim 1, the cited reference must teach or suggest the relevant limitations of claim 1. For at least the reasons described above, the current rejection has failed to do so; Applicants respectfully submit the current rejection should be withdrawn. Applicants further submit claim 1 is allowable, and independent claims 9, 15, and 21, containing similar limitations, are allowable as well. Claims 2-8, 10-14, 16-20, and 22-28 are allowable at least for depending from an allowable base claim.

It is believed that this Response places the application in condition for allowance, and early favorable consideration of this Response is earnestly solicited.

If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the telephone number listed below.

The Office is hereby authorized to charge any fees, or credit any overpayments, to Deposit Account No. **11-0600**.

Respectfully submitted,

KENYON & KENYON LLP

Dated: June 23, 2008

By: /Sumit Bhattacharya/
Sumit Bhattacharya
(Reg. No. 51,469)
Attorneys for Intel Corporation

KENYON & KENYON LLP
333 West San Carlos St.
San Jose, CA 95110

Telephone: (408) 975-7500
Facsimile: (408) 975-7501